MULTIMEDIA UNIVERSITY®

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/2019

ECE1026 ALGORITHMS AND DATA STRUCTURES

(All Sections / Groups)

11 MARCH 2019
9:00 A.M – 11:00 A.M.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 8 printed pages including the cover page with 4 questions only.

2. Attempt all FOUR (4) questions. All questions carry equal marks and the distribution of marks is given in the questions.

3. Please print all your answers in the answer booklet provided.

4. State all assumptions clearly.

Question 1

(a) A finite state machine is shown in Figure Q1(a).

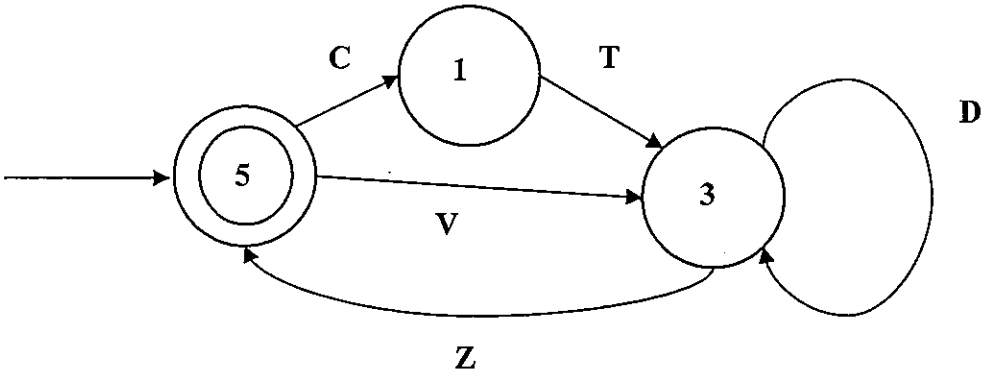


Figure Q1(a)

Based on the figure, determine,

(i) The Regular Expression (RE) for the finite state machine.

[5 marks]

(ii) The set of all states (Q), the set of all input symbols ($\sum$), the initial state (q_0), and the set of final states (F) for the finite state machine.

[5 marks]

(iii) Two accepted words for the finite state machine.

[2 marks]

(b) Draw a finite state machine for RE = 2* (6+ 8 | 4).

[5 marks]

Continued...

(c) Given the Turing Machine in Figure Q1(c), write down the content of the tape and position of the read/write head at the end of each execution cycle until the machine stops. Assume that the read/write head is pointing to the first left most non-blank square.

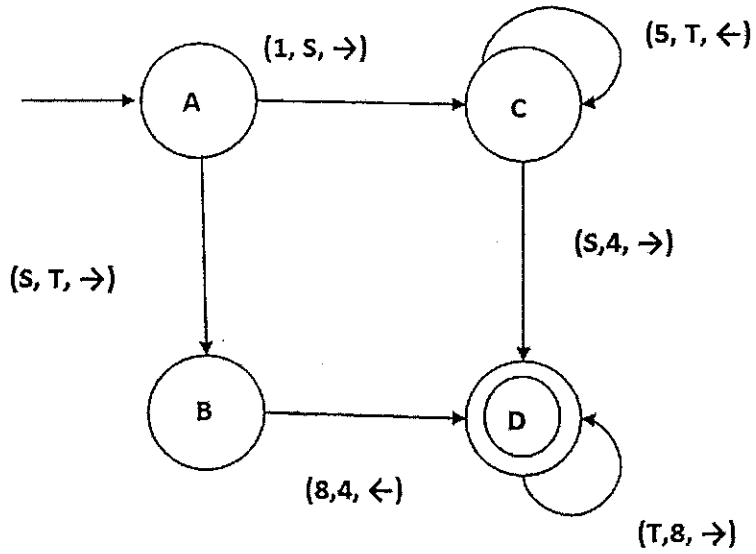
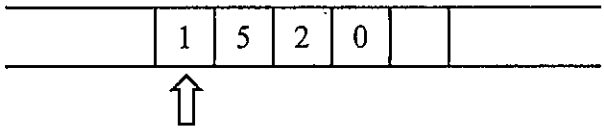


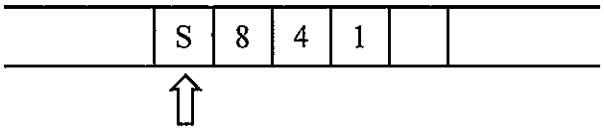
Figure Q1(c)

(i) Tape X



[4 marks]

(ii) Tape Y



[4 marks]

Continued...

Question 2

(a) Consider a partial linked list C program as shown in Figure Q2(a) below.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

//Answer for Part (a)(i) - struct Mark definition

struct Mark *addMark(char [],int, int, int);

int main(){
    struct Mark *head=NULL, *current=NULL;
    head = current = addMark= ("",0,0,0);

   //Answer for Part (a)(iii) - store data

   //Answer for Part (a)(iv) - display total marks and highest mark

    return(0);
}

//Answer for Part (a)(ii) - function addMark definition
```

Figure Q2(a)

(i) Write C code to define a suitable linked list node structure **`Mark`** to represent the assessment marks shown in Table Q2.

[3 marks]

(ii) Write a function definition for **`addMark`** which creates and initializes a new **`Mark`** node.

[4 marks]

(iii) Write C code to store assessment marks from Table Q2 in a linked list.

[4 marks]

Table Q2: Assessment Marks

Name	Test	Lab	Assignment
Sofea	18	21	17
Umar	9	11	12
Mary	30	28	25
Paul	27	23	22

Continued...

(iv) Write C code to calculate the **total marks** for each student and determine who gets the **highest marks** from data stored in the linked list. Display the results on the monitor screen. A sample output is as shown in Figure Q2(b).

[5 marks]

```
Name        Total Marks
Sofea       56
Umar        32
Mary        83
Paul        72

Mary has achieved the highest marks.
```

Figure Q2(b): Sample Output

(b) Study the weighted graph in Figure Q2(c).

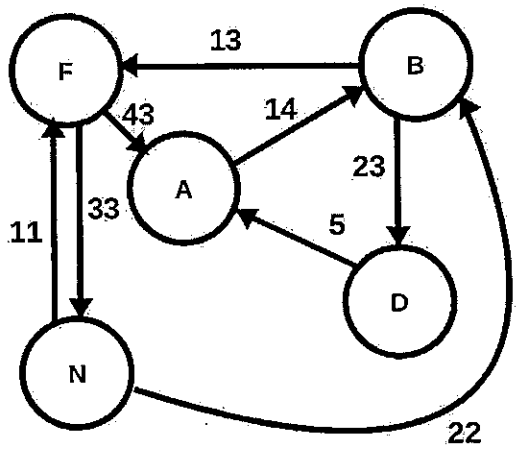


Figure Q2(c): Weighted Graph

(i) How many unique cycles are there in this graph?

[2 marks]

(ii) Write a simple path that includes all vertices.

[2 marks]

(iii) Suppose a global array is declared and initialized as follows:

```
char node[5]={'A', 'B', 'D', 'F', 'N'};
```

Write a C declaration code that will create an adjacency matrix (array) representation of this graph.

[5 marks]

Continued...

Question 3

(a) Define the term, *recursion*. Discuss the advantages and disadvantages of using recursive function.

[6 marks]

(b) Study the following formula:

```
f(x,y) = 1                  if y = 0;
f(x,y) = x * f(x,y-1)       if y > 0;
f(x,y) = 1/f(x,-y)          if y < 0.
```

(i) Write a direct *recursive* function in C programming language based on the formula above.

[6 marks]

(ii) Write a *complete* C program to test the recursive function you wrote in Part (i). You need to include the recursive function in your answer, showing its position relative to the other parts of the program. Prompt the user for the values of **x** and **y** and print the result to the monitor as shown in the example below.

```
Enter x and y values: 2 -3
f(2,-3) = 0.125000
```

[6 marks]

(c) The *N*-queen problem is to place *N* queens on an $N \times N$ chessboard in such a way that no Queen is being attacked by another Queen either diagonally, horizontally or vertically.

(i) Determine the three steps of the backtracking algorithm used to solve the *N*-Queen problem.

[3 marks]

(ii) Figure Q3(c) shows a 5×5 chessboard with two Queens. Using the algorithm described in Part (i), complete the solution for the 5-Queen problem by placing the other 4 Queens onto the chessboard. Reproduce Figure Q3(c) in your answer, showing the positions of the 5 Queens.

[4 marks]

Q				

Figure Q3(c)

Continued...

Question 4

(a) The incomplete C program as shown in Figure Q4(a) implements a search for ISBN of books.

[10 marks]

```
#include <stdio.h>
#include <string.h>
#define SIZE 5

struct Book {
   int isbn;
   char title[50];
   char author[30];
};

//Function definition for Q4(a)(i)
//Function definition for Q4(a)(ii)
//Function definition for Q4(a)(iii)

int main() {
   int user_ISBN, result;
   struct Book top_sellers[SIZE] = {
      {1528, "Becoming", "Michelle Obama"},
      {2564, "The Hobbit", "JRR Tolkien"},
      {3432, "To Kill a Mockingbird", "Harper Lee"},
      {4563, "Harry Potter and the Goblet of Fire", "JK Rowling"},
      {8948, "Everyday Super Food", "Jamie Oliver"}};

   printf("Enter ISBN: ");
   scanf("%d", &user_ISBN);

   result = sequentialSearch(user_ISBN, top_sellers);
   displayResult(result, top_sellers);

   result = binarySearch(user_ISBN, top_sellers);
   displayResult(result, top_sellers);

   return 0;
}
```

Figure Q4(a)

(i) Write the C function **`sequentialSearch`** that performs *sequential search* to search for the ISBN in the array of records. The function takes in the **`user_ISBN`** and array of **`Book`** structures. If a match is found, the function returns the index of the matching element, otherwise it returns **`-1`**.

[6 marks]

(ii) Write the C function **`binarySearch`** that performs *binary search* to search for the ISBN in the array of records. The function takes in the **`user_ISBN`** and array of **`Book`** structures. If a match is found, the function returns the index of the matching element, otherwise it returns **`-1`**. Use *iterative* method with a **`while`** loop.

[7 marks]

Continued...

(iii) Write the C function **`displayResult`** that prints out the details (i.e. **`isbn`**, **`titile`**, **`author`**) of the book. The function takes in the **`result`** and array of **`Book`** structures. If there are no results, display "**`Not Found`**" instead.

[3 marks]

(b) Sort by hand the word "**`DEMOLISH`**" in *ascending* order using the following sorting algorithms. Show all steps involved.

(i) Selection sort

[3 marks]

(ii) Insertion sort

[3 marks]

(iii) Quick sort (the middle element as pivot)

[3 marks]

End